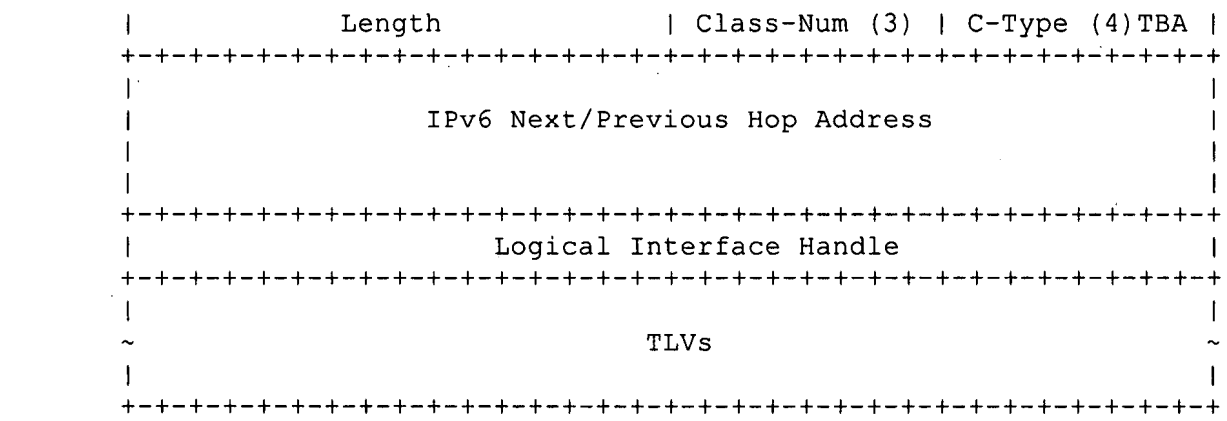


See [RFC2205] for a description of hop address and handle fields. See [GMPLS-SIG] for a description of parameters and encoding of TLVs.

8.1.2. Procedures

An IF_ID RSVP_HOP object is used in place of previously defined RSVP_HOP objects. It is used on links where there is not a one-to-one association of a control channel to a data channel, see [GMPLS-SIG]. The Hop Address and Logical Interface Handle fields are used per standard RSVP [RFC2205].

TLVs are used to identify the data channel(s) associated with the LSP. For a unidirectional LSP, a forward channel MUST be indicated. For a bidirectional LSP that uses bundled links, a reverse channel MUST be indicated. Data channels are specified from the view point of the sender of the Path message. The IF_ID RSVP_HOP object SHOULD NOT be used when no TLVs are needed.

A node receiving one or more TLVs in a Path message saves their values and returns them in the HOP objects of subsequent Resv messages sent to the node that originated the TLVs.

As with [MPLS-TE], the node originating an IF_ID object must ensure that the selected outgoing interface is consistent with the outgoing ERO. A node that receives an IF_ID object SHOULD check whether the information carried in this object is consistent with the information carried in a received ERO, and if not it MUST send a PathErr with the error code "Routing Error" and error value of "Bad Explicit Route

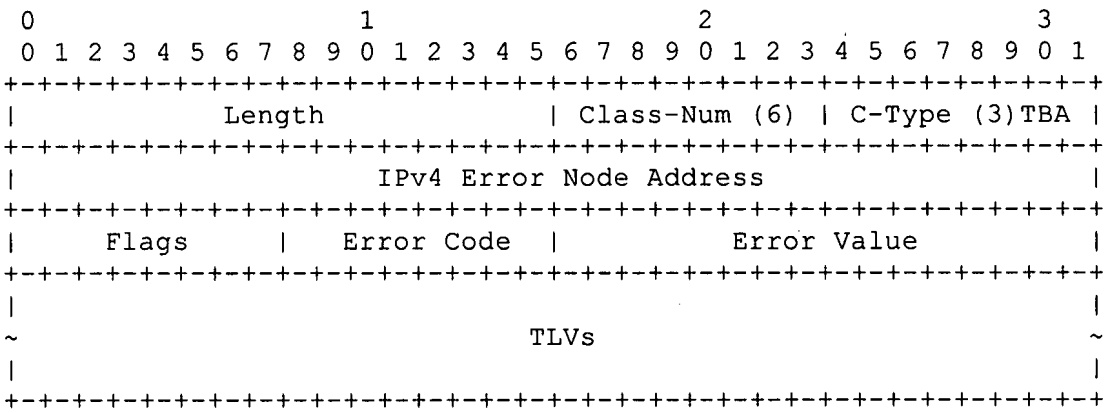
Object" toward the sender.

8.2. Errored Interface Identification

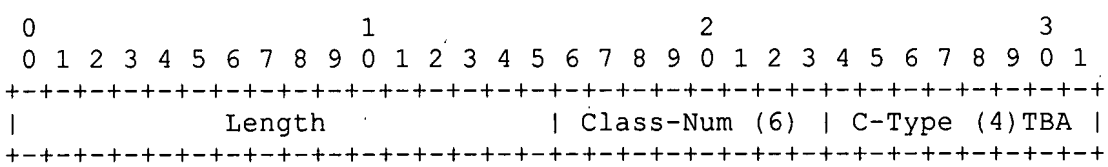
There are cases where it is useful to indicate a specific interface associated with an error. To support these cases the IF_ID ERROR_SPEC Objects are defined.

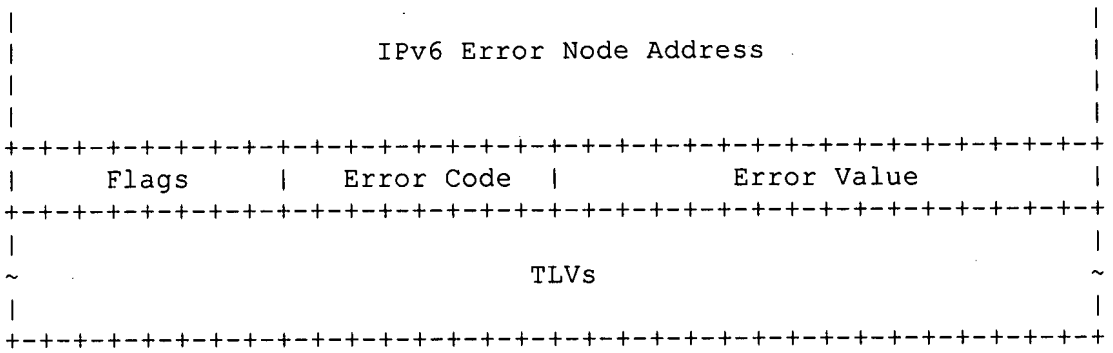
8.2.1. IF_ID ERROR_SPEC Objects

The format of the IPv4 IF_ID ERROR_SPEC Object is:



The format of the IPv6 IF_ID ERROR_SPEC Object is:





See [RFC2205] for a description of address, flags, error code and error value fields. See [GMPLS-SIG] for a description of parameters and encoding of TLVs.n

8.2.2. Procedures

Nodes wishing to indicate that an error is related to a specific interface SHOULD use the appropriate IF_ID ERROR_SPEC Object in the corresponding PathErr or ResvErr message. IF_ID ERROR_SPEC Objects SHOULD be generated and processed as any other ERROR_SPEC Object, see [RFC2205].

9. Fault Handling

The handling of two types of control communication faults is described in this section. The first, referred to as nodal faults, relates to the case where a node losses its control state (e.g., after a restart) but does not loose its data forwarding state. In the second, referred to as control channel faults, relates to the case where control communication is lost between two nodes. The handling of both faults is supported by the Restart_Cap object defined below and require the use of Hello messages.

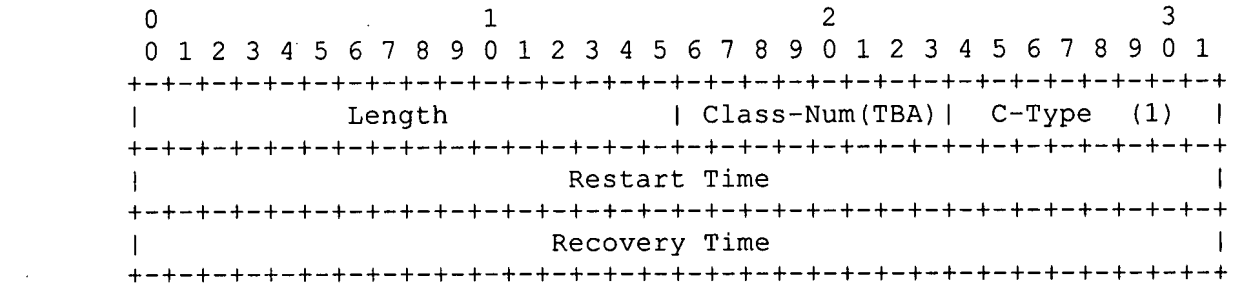
Note, the Restart_Cap object MUST NOT be sent when there is no mechanism to detect data channel failures independent of control channel failures.

Please note this section is derived from [PAN-RESTART].

9.1. Restart_Cap Object

The Restart_Cap Object is carried in Hello messages.

The format of the Restart_Cap Object is:



Restart Time: 32 bits

Restart Time is measured in milliseconds. Restart Time SHOULD be set to the sum of the time it takes the sender of the object to restart its RSVP-TE component (to the point where it can exchange RSVP Hello with its neighbors) and the communication channel that is used for RSVP communication.

Recovery Time: 32 bits

The period of time, in milliseconds, that the sender desires for the recipient to resyncronize RSVP and MPLS forwarding state with the sender after the re-establishment of Hello synchronization. A value of zero (0) indicates that MPLS forwarding state was not preserved across a particular reboot. A value of 0xffffffff indicates that resynchronization may occur at a rate selected by the receiver.

9.2. Processing of Restart_Cap Object

Nodes supporting state recovery advertise this capability by carrying the Restart_Cap object in Hello messages. Such nodes MUST include the Restart_Cap object in all Hello messages. (Note that this includes Hello messages containing ACK objects.) Usage of the special case Recovery Time values is described in greater detail below.

When a node receives a Hello message with the Restart_Cap object, it SHOULD record the values of the parameters received.

9.3. Modification to Hello Processing to Support State Recovery

When a node determines that RSVP communication with a neighbor has been lost, and the node previously learned that the neighbor supports state recovery, the node SHOULD wait at least the amount of time indicated by the Restart Time indicated by the neighbor before invoking procedures related to communication loss. A node MAY wait longer based on local policy or configuration information.

During this waiting period, all Hello messages MUST be sent with a Dst_Instance value set to zero (0), and Src_Instance should be unchanged. While waiting, the node SHOULD also preserve the RSVP and MPLS forwarding state for (already) established LSPs that traverse the link(s) between the node and the neighbor. In a sense with respect to established LSPs the node behaves as if it continues to receive periodic RSVP refresh messages from the neighbor. The node

MAY clear RSVP and forwarding state for the LSPs that are in the process of being established when their refresh timers expire. Refreshing of Resv and Path state SHOULD be suppressed during this waiting period.

During this waiting period, the node MAY inform other nodes of the communication loss via a PathErr and/or upstream Notify message with "Control Channel Degraded State" indication. If such notification has been sent, then upon restoration of the control channel the node MUST inform other nodes of the restoration via a PathErr and/or upstream Notify message with "Control Channel Active State" indication. (Specific error codes are to be assigned IANA.)

When a new Hello message is received from the neighbor, the node must determine if the fault was limited to the control channel or was a nodal fault. This determination is based on the Src_Instance received from the neighbor. If the value is different than the value that was received from the neighbor prior to the fault, then the neighbor should be treated as if it has restarted. Otherwise, the the fault was limited control channel. Procedures for handling each case are described below.

9.4. Control Channel Faults

In the case of control channel faults, the node SHOULD refresh all state shared with the neighbor. Summary Refreshes [RSVP-RR] with the ACK_Desired flag set SHOULD be used, if supported. Note that if a large number of messages are need, some pacing should be applied.

All state SHOULD be refreshed within the Recovery time advertised by the neighbor.

9.5. Nodal Faults

Recovering from nodal faults uses one new object and other existing protocol messages and objects.

9.5.1. Recovery Label

The Recovery_Label object is used during the nodal fault recovery process. The format of a Recovery_Label object is identical to a generalized label. A Recovery_Label object uses Class-Number TBA (of form 0bbbbbbb) and the C-Type of the label being suggested.

9.5.2. Procedures for the Restarting node

After a node restarts its control plane, a node that supports state recovery SHOULD check whether it was able to preserve its MPLS forwarding state. If no forwarding state from prior to the restart was preserved, then the node MUST set the Recovery Time to 0 in the Hello message the node sends to its neighbors.

If the forwarding state was preserved, then the node initiates the state recovery process. The period during which a node is prepared to support the recovery process is referred to as the Recovery Period. The total duration of the Recovery Period is advertised by the recovering node in the Recovery Time parameter of the Restart_Cap object. The Recovery Time MUST be set to the duration of the Recovery Period in all Hello messages sent during the Recovery Period. A Recovery Time value of 0xffffffff indicates that the Recovery Period is effectively infinite. State that is not resynchronized during the Recovery Period SHOULD be removed at the end of the Period.

Note that if during Hello synchronization the restarting node determines that a neighbor does not support state recovery, and the restarting node maintains its MPLS forwarding state on a per neighbor basis, the restarting node should immediately consider the Recovery Period with that neighbor completed. Note forwarding state can be considered to be maintained on a per neighbor basis when per interface labels are used on point-to-point interfaces.

When a node receives a Path message during the Recovery Period, the node first checks if it has an RSVP state associated with the

message. If the state is found, then the node handles this message according to previously defined procedures.

If the RSVP state is not found, and the message does not carry a Recovery_Label object, the node treats this as a setup for a new LSP, and handles it according to previously defined procedures.

If the RSVP state is not found, and the message carries a Recovery_Label object, the node searches its MPLS forwarding table (the one that was preserved across the restart) for an entry whose incoming interface matches the Path message and whose incoming label is equal to the label carried in the Recovery_Label object.

If the MPLS forwarding table entry is not found, the node treats this as a setup for a new LSP, and handles it according to previously defined procedures.

If the MPLS forwarding table entry is found, the appropriate RSVP

state is created, the entry is bound to the LSP associated with the message, and related forwarding state should be considered as valid and refreshed. Normal Path message processing should also be conducted. When sending the corresponding outgoing Path message the node SHOULD include a Suggested_Label object with a label value matching the outgoing label from the now restored forwarding entry. The outgoing interface SHOULD also be selected based on the forwarding entry.

Additionally, for bidirectional LSPs, the node extracts the label from the UPSTREAM_LABEL object carried in the received Path message, and searches its MPLS forwarding table for an entry whose outgoing label is equal to the label carried in the object (in the case of link bundling, this may also involved first identifying the appropriate incoming component link).

If the MPLS forwarding table entry is not found, the node treats this as a setup for a new LSP, and handles it according to previously defined procedures.

If the MPLS forwarding table entry is found, the entry is bound to the LSP associated with the Path message, and the entry should be considered to be resyncronized. In addition, if the node is not the tail-end of the LSP, the corresponding outgoing Path messages is sent with the incoming label from that entry carried in the UPSTREAM_LABEL object.

During the Recovery Period, Resv messages are processed normally with two exceptions. In the case that a forwarding entry is recovered, no new label or resource allocation is required while processing the Resv message. The second exception applies only if the Recovery Time

is not 0xffffffff. In this case, ResvErr messages SHOULD NOT be generated when a Resv message with no matching Path state is received. In this case the Resv message SHOULD just be silently discarded.

9.5.3. Procedures for the Neighbor of a Restarting node

The following specifies the procedures that apply when the node reestablishes communication with the neighbor's control plane within the Restart Time, the node determines (using the procedures defined in Section 5 of [RSVP-TE]) that the neighbor's control plane has restarted, and the neighbor was able to preserve its forwarding state across the restart (as was indicated by a non-zero Recovery Time carried in the Restart_Cap object of the RSVP Hello messages received from the neighbor).

Upon detecting a restart with a neighbor that supports state recovery, a node SHOULD refresh all Path state shared with that neighbor. The outgoing Path messages MUST include the Recovery_Label object containing the label value received in the most recently received corresponding Resv message. All Path state SHOULD be refreshed within approximately 1/2 of the Recovery time advertised by the restarted neighbor. If there are many LSP's going through the restarting node, the neighbor node should avoid sending Path messages in a short time interval, as to avoid unnecessary stressing the restarting node's CPU. Instead, it should spread the messages across 1/2 the Recovery Time interval.

After detecting a restart of a neighbor that supports state recovery, all Resv state shared with the restarting node MUST NOT be refreshed until a corresponding Path message is received. This requires suppression of normal Resv and Summary Refresh processing to the neighbor during the Recovery Time advertised by the restarted neighbor. As soon as a corresponding Path message is received a Resv message SHOULD be generated and normal state processing SHOULD be re-enabled.

10. RSVP Message Formats and Handling

This message summarizes RSVP message formats and handling as modified by GMPLS.

10.1. RSVP Message Formats

This section presents the RSVP message related formats as modified by this document. Where they differ, formats for unidirectional LSPs

are presented separately from bidirectional LSPs. Unmodified formats are not listed. Again, MESSAGE_ID and related objects are defined in [RSVP-RR].

The format of a Path message is as follows:

```
<Path Message> ::=      <Common Header> [ <INTEGRITY> ]
                        [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                        [ <MESSAGE_ID> ]
                        <SESSION> <RSVP_HOP>
                        <TIME_VALUES>
                        [ <EXPLICIT_ROUTE> ]
                        <LABEL_REQUEST>
                        [ <PROTECTION> ]
                        [ <LABEL_SET> ... ]
                        [ <SESSION_ATTRIBUTE> ]
                        [ <NOTIFY_REQUEST> ]
                        [ <ADMIN_STATUS> ]
                        [ <POLICY_DATA> ... ]
                        <sender descriptor>
```

The format of the sender description for unidirectional LSPs is:

```
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                        [ <ADSPEC> ]
                        [ <RECORD_ROUTE> ]
                        [ <SUGGESTED_LABEL> ]
                        [ <RECOVERY_LABEL> ]
```

The format of the sender description for bidirectional LSPs is:

```
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                        [ <ADSPEC> ]
                        [ <RECORD_ROUTE> ]
                        [ <SUGGESTED_LABEL> ]
                        [ <RECOVERY_LABEL> ]
                        <UPSTREAM_LABEL>
```

The format of a PathErr message is as follows:

```
<PathErr Message> ::=    <Common Header> [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION> <ERROR_SPEC>
                         [ <ACCEPTABLE_LABEL_SET> ... ]
                         [ <POLICY_DATA> ... ]
                         <sender descriptor>
```


The format of a Resv message is as follows:

```
<Resv Message> ::=       <Common Header> [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION> <RSVP_HOP>
                         <TIME_VALUES>
                         [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                         [ <NOTIFY_REQUEST> ]
                         [ <ADMIN_STATUS> ]
                         [ <POLICY_DATA> ... ]
                         <STYLE> <flow descriptor list>
```

<flow descriptor list> is not modified by this document.

The format of a ResvErr message is as follows:

```
<ResvErr Message> ::=    <Common Header> [ <INTEGRITY> ]
                         [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
                         [ <MESSAGE_ID> ]
                         <SESSION> <RSVP_HOP>
                         <ERROR_SPEC> [ <SCOPE> ]
                         [ <ACCEPTABLE_LABEL_SET> ... ]
                         [ <POLICY_DATA> ... ]
                         <STYLE> <error flow descriptor>
```

The modified Hello message format is:

```
<Hello Message> ::= <Common Header> [ <INTEGRITY> ] <HELLO>
                    [ <RESTART_CAP> ]
```

10.2. Addressing Path and PathTear Messages

RSVP was designed to handle dynamic (non-explicit) path changes and non RSVP hops along the path. To this end, the Path and PathTear messages carry the destination address of the session in the IP header. In generalized signaling, routes are usually explicitly signaled. Further, hops that cannot allocate labels cannot exist in the path of an LSP. A further difference with traditional RSVP is that at times, an RSVP message may travel out of band with respect to an LSP's data channel.

When a node is sending a Path or PathTear message to a node that it knows to be adjacent at the data plane (i.e. along the path of the LSP) it SHOULD address the message directly to an address associated with the adjacent node's control plane. In this case the router-alert option SHOULD not be included.

11. Acknowledgments

This draft is the work of numerous authors and consists of a composition of a number of previous drafts in this area. A list of the drafts from which material and ideas were incorporated follows:

draft-saha-rsvp-optical-signaling-00.txt
draft-lang-mpls-rsvp-oxc-00.txt
draft-kompella-mpls-optical-00.txt
draft-fan-mpls-lambda-signaling-00.txt
draft-pan-rsvp-te-restart-01.txt

Valuable comments and input were received from a number of people, including Igor Bryskin, Adrian Farrel and Dimitrios Pendarakis. Portions of Section 4 are based on suggestions and text proposed by Adrian Farrel.

12. Security Considerations

The transmission of notify messages using IP in IP, breaks RSVP's hop-by-hop integrity and authentication model. Fortunately, such usage mirrors the IP end-to-end model. In the case where RSVP is generating end-to-end messages and integrity and/or authentication are desired, the standard IPSEC based integrity and authentication methods SHOULD be used.

This draft introduces no other new security considerations to [RSVP-TE].

13. IANA Considerations

IANA assigns values to RSVP protocol parameters. Within the current document multiple objects are defined. Each of these objects contain

C-Types. This section defines the rules for the assignment of the related C-Type values. This section uses the terminology of BCP 26 "Guidelines for Writing an IANA Considerations Section in RFCs" [BCP26].

As per [RFC2205], C-Type is an 8-bit number that identifies the function of an object. There are no range restrictions. All possible values except zero are available for assignment.

The assignment of C-Type values of the objects defined in this document fall into three categories. The first category inherit C-Types from the Label object, i.e., object class number 16 [RSVP-TE]. IANA is requested to institute a policy whereby all C-Type values

assign for the Label object are also assigned for the following objects:

- o Suggested_Label (Class-Num TBA)
- o Upstream_Label (Class-Num TBA)
- o Recovery_Label (Class-Num TBA)

The second category of objects follow independent policies. Specifically, following the policies outlined in [BCP26], C-Type values in the range 0x00 - 0x3F are allocated through an IETF Consensus action, values in the range 00x40 - 0x5F are allocated as First Come First Served, and values in the range 0x60 - 0x7F are reserved for Private Use. This policy applies to the following objects.

- o Label_Set (Class-Num TBA)
- o Notify_Request (Class-Num TBA)
- o Protection (Class-Num TBA)
- o Admin Status (Class-Num TBA)
- o Restart_Cap (Class-Num TBA)

The assignment of C-Type values for the remaining object, the Acceptable_Label_Set object, follows the assignment of C-Type values of the Label_Set object. IANA is requested to institute a policy whereby all C-Type values assigned for the Label_Set object are also assigned for the Acceptable_Label_Set object.

14. References

[BCP26] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[MPLS-BUNDLE] Kompella, K., Rekhter, Y., and Berger, L., "Link Bundling in MPLS Traffic Engineering", Internet Draft, draft-kompella-mpls-bundle-05.txt, Feb., 2001.

[MPLS-HIERARCHY] Kompella, K., and Rekhter, Y., "LSP Hierarchy with

MPLS TE", Internet Draft,
draft-ietf-mpls-lsp-hierarchy-02.txt, Feb., 2001.

[MPLS-UNNUM] Kompella, K., Rekhter, Y., "Signalling Unnumbered Links in RSVP-TE", Internet Draft, draft-ietf-mpls-rsvp-unnum-02.txt, August 2001

[GMPLS-LDP] Ashwood-Smith, P. et al, "Generalized MPLS Signaling - CR-LDP Extensions", Internet Draft, draft-ietf-mpls-generalized-cr-ldp-01.txt, February 2001.

[GMPLS-SIG] Ashwood-Smith, P. et al, "Generalized MPLS - Signaling Functional Description", Internet Draft, draft-ietf-mpls-generalized-signaling-02.txt, February 2001.

[PAN-RESTART] Pan, P, et al, "Graceful Restart Mechanism for RSVP-TE", Internet Draft, draft-pan-rsvp-te-restart-01.txt, July 2001.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119.

[RFC2205] Braden, R. Ed. et al, "Resource ReserVation Protocol -- Version 1 Functional Specification", RFC 2205, September 1997.

[RSVP-TE] Awduche, D.O. et al, "RSVP-TE: Extensions to RSVP for LSP Tunnels," Internet Draft, draft-ietf-mpls-rsvp-lsp-tunnel-08.txt, February 2001.

[RSVP-RR] Berger, L. et al, "RFC 2961: RSVP Refresh Overhead Reduction Extensions", RFC2961.

15. Authors' Addresses

Peter Ashwood-Smith
Nortel Networks Corp.
P.O. Box 3511 Station C,
Ottawa, ON K1Y 4H7
Canada
Phone: +1 613 763 4534
Email: petera@nortelnetworks.com

Ayan Banerjee
Calient Networks
5853 Rue Ferrari

San Jose, CA 95138
Phone: +1 408 972-3645
Email: abanerjee@calient.net

Lou Berger
Movaz Networks, Inc.
7926 Jones Branch Drive
Suite 615
McLean VA, 22102
Phone: +1 703 847-1801
Email: lberger@movaz.com

Greg Bernstein
Ciena Corporation
10480 Ridgeview Court
Cupertino, CA 94014
Phone: +1 408 366 4713
Email: greg@ciena.com

John Drake
Calient Networks
5853 Rue Ferrari
San Jose, CA 95138
Phone: +1 408 972 3720
Email: jdrake@calient.net

Yanhe Fan
Axiowave Networks, Inc.
200 Nickerson Road
Marlborough, MA 01752
Phone: +1 774 348 4627
Email: yfan@axiowave.com

Kireeti Kompella
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
Email: kireeti@juniper.net

Jonathan P. Lang
Calient Networks
25 Castilian

Goleta, CA 93117
Email: jplang@calient.net

Fong Liaw
Zaffire Inc.
2630 Orchard Parkway,
San Jose, CA 95134
Email: fliaw@zaffire.com

Eric Mannie
EBONE
Terhulpsesteenweg 6A
1560 Hoeilaart - Belgium
Phone: +32 2 658 56 52
Mobile: +32 496 58 56 52
Fax: +32 2 658 51 18
Email: eric.mannie@ebone.com

Ping Pan
Juniper Networks
1194 N.Mathilda Ave
Sunnyvale, CA 94089
Email: pingpan@juniper.net

Bala Rajagopalan
Tellium, Inc.
2 Crescent Place
P.O. Box 901
Oceanport, NJ 07757-0901
Phone: +1 732 923 4237
Fax: +1 732 923 9804
Email: braja@tellium.com

Yakov Rekhter
Juniper Networks, Inc.
Email: yakov@juniper.net

Debanjan Saha
Tellium Optical Systems
2 Crescent Place
Oceanport, NJ 07757-0901
Phone: +1 732 923 4264
Fax: +1 732 923 9804
Email: dsaha@tellium.com

Vishal Sharma
Metanoia, Inc.
335 Elan Village Lane, Unit 203

San Jose, CA 95134-2539
Phone: +1 408-943-1794
Email: v.sharma@ieee.org

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Voice: +1 978 244 8143
Email: swallow@cisco.com

Z. Bo Tang
Tellium, Inc.
2 Crescent Place
P.O. Box 901
Oceanport, NJ 07757-0901
Phone: +1 732 923 4231
Fax: +1 732 923 9804
Email: btang@tellium.com

Generated on: Mon Oct 1 01:20:20 EDT 2001

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	(re$start re$set) with reservation same (fail$3 or error)	US-PGPUB; USPAT; EPO	SAME	ON	2007/06/14 08:32
L2	6362178	@ad<"20020920" or @prad<"20020920"	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 08:33
L3	61512	resource near5 (reservat$3 allocat$3 manag$3 management)	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 08:33
L4	755	(re-set$3 re$set$3) with (path link) and L3 and L2	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 08:57
L6	37	4 and mpls	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 08:34
L7	38	4 and (mpls or label near switch$3)	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 08:35
L8	10	reservation with (re-set$3 re$set$3 re$fresh retry3 re$tries) same (path flow link) same (fail$3) and L3 and L2	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 09:00
L9	10	("6515966").URPN.	USPAT	OR	ON	2007/06/14 09:27
S1	6361962	@ad<"20020920" or @prad<"20020920"	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 10:48
S2	61349	resource near5 (reservat$3 allocat$3 manag$3 management)	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 14:50
S3	3858	label near switch$3	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 10:50
S4	739	S1 and S2 and S3	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 10:50
S5	19146	709/223-226,229,232-234.ccls.	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 10:55
S6	95	S4 and S5	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 10:56
S7	3	(re-set$3 re$set$3) with (path link) and S6	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 10:59

EAST Search History

S8	32	(re-set$3 re$set$3) with (path link) and S4	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 13:45
S9	0	(re-set$3 re$set$3) with (path link) same period$8 and S4	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 13:46
S10	0	((re-set$3 re$set$3) with (path link) same period$8)and S4	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 11:02
S11	755	(re-set$3 re$set$3) with (path link) and S2 and S1	US-PGPUB; USPAT; EPO	OR	ON	2007/06/14 08:33
S12	35699	S1 and S2	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 15:05
S13	1	("20040073650").PN.	US-PGPUB; USPAT; EPO	OR	OFF	2007/06/13 15:36
S14	36	reservation with fail$3 same (count threshold)	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 15:37
S15	198	reservation with fail$3 same (count number threshold)	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 15:38
S16	39	S12 and S15	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 15:49
S17	1	("2002374292").PN.	US-PGPUB; USPAT; EPO; JPO	OR	OFF	2007/06/13 15:43
S18	3	S16 and mpls	US-PGPUB; USPAT; EPO	OR	ON	2007/06/13 16:15
S19	1	S12 and reservation (fail$3 error) (count number threshold)	US-PGPUB; USPAT; EPO	SAME	ON	2007/06/14 08:26
S20	194	S12 and (reservation (fail$3 or error) (count or number or threshold))	US-PGPUB; USPAT; EPO	SAME	ON	2007/06/13 16:18
S21	47	S12 and (reservation (fail$3 or error) (count or threshold))	US-PGPUB; USPAT; EPO	SAME	ON	2007/06/13 16:20
S22	32	S21 and bandwidth	US-PGPUB; USPAT; EPO	SAME	ON	2007/06/13 16:28

EAST Search History

S23	16	"664635"	US-PGPUB; USPAT; EPO	SAME	ON	2007/06/13 16:28
S24	0	("2004/0006218").URPN.	USPAT	OR	ON	2007/06/13 16:52
S25	2	reset3 bandwidth reservation	USPAT	WITH	ON	2007/06/13 16:53